

Docket No. BOC9-2000-0079(214)

**A METHOD AND APPARATUS FOR PROVIDING DYNAMIC WORKLOAD
TRANSITION DURING WORKLOAD SIMULATION ON E-BUSINESS APPLICATION
SERVER**

Inventor(s):
Romelia Flores
Leonard S. Hand
Philip E. Reed

International Business Machines Corporation

IBM DOCKET BOC9-2000-0079
IBM DISCLOSURE BOC8-2000-0095

EXPRESS MAIL LABEL NO.: EL920516481US

BACKGROUND OF THE INVENTION

Technical Field

The present invention relates to the field of e-business, and more particularly, to a method and apparatus for providing dynamic workload transition during workload simulation on an e-business application server.

Description of the Related Art

The pervasiveness of the Internet has allowed companies to exploit electronic communications to engage in what is commonly known as e-business activities with their customers. E-business involves conducting business on the Internet and not only includes buying and selling goods and services, but also includes servicing customers and collaborating with trading or business partners. To accommodate this vast range of activities, companies engaging in e-business have to ensure that their systems operate optimally by implementing a traffic flow management policy.

A traffic flow management policy is a set of rules which dictate how workloads should be handled by a system and its subsystems. A workload is a task or group of tasks that require system resources for processing. A work request can be used to initiate processing of a workload. An effective traffic flow management policy must assign and distribute tasks associated with a workload to various processing elements within the system and its subsystems. Moreover, to ensure optimal performance, the

traffic flow management policy must constantly monitor system conditions and dynamically adjust the workload accordingly. Dynamic adjustment can include redistributing and rescheduling the tasks associated with a workload for processing.

E-business systems present a unique set of challenges to providing an effective traffic flow management policy. In e-business systems, legitimate overload conditions occur frequently and are generally unpredictable. A legitimate overload condition may result from several load factors which can include an increase in user demand, a partial website outage, and even the elimination of previous bottlenecks. For example, an increased user demand can occur when there are special events, promotions, and marketing campaigns facilitated by the website. Furthermore, when overload occurs, system performance is significantly degraded and this results in diminished capacity. From an economic perspective, the diminished capacity results in lost opportunity which translates directly to lost revenue.

Given the unique challenges presented by e-business systems, what is needed is an efficient method and system for applying resource management policies to transition different workloads in order to provide a robust e-business optimally operating system.

SUMMARY OF THE INVENTION

The invention provides a method and system for dynamic workload transition in an application server for an e-business system. The method can include the steps of detecting an overload condition in the e-business system and executing a lighter workload upon detecting the system overload condition. Executing a lighter workload reduces system resources and prevents overload. Whenever adequate system resources become available, a request for the original workload or any new workload can be processed as normal. The step of detecting an overload condition can include monitoring system parameters, such as CPU utilization, disk I/O and memory utilization in the e-business system. The system parameters can be monitored to determine when an overload condition occurs in the e-business system.

In another aspect of the invention, the method can include the steps of diminishing the processing load of a first workload that causes a system overload condition and transitioning to a second lighter workload. Since the second lighter workload consumes less system resources than the first workload, it will add to the system overload condition. Subsequent requests that are made for the original workload will result in the processing of the second lighter workload until the system status indicates that the system is not overloaded. Once the system status returns to a normal condition where the overload condition does not exist, requests for the original workload will result in the first workload being processed. The method can further include the step of analyzing system parameters to determine causes for the system

overload condition. The system parameters can include, but are not limited to, CPU utilization, disk I/O, memory utilization, and page ins. The method can also include the step of reporting the system parameters to a workload driver.

The invention also provides a method for dynamic workload transition in an application server for an e-business system. The method can include processing a workload assigned to a workload driver. The e-business system can be monitored to detect when overload conditions occur. Based on the results of the monitoring, a lighter workload can be processed whenever the workload driver detects a system overload condition caused by the processed workload. Upon a request to process the original workload and available adequate processing resources, the workload driver can transition to the originally processed workload.

A system for providing dynamic workload transition in an e-business system is also disclosed. The system can include an application server for processing work requests and for processing workloads identified by the work requests. A workload driver can be utilized for handling workload management of the application server. The handling can include reducing processing resources necessary for processing a currently processed workload when an overload condition exists by initiating the processing of a workload which has a lighter load and requires less system resources. A status driver can report system data to the workload driver, and the reported system data can provide information necessary for determining the existence of the overload condition.

BRIEF DESCRIPTION OF THE DRAWINGS

There are presently shown in the drawings embodiments which are presently preferred, it being understood, however, that the invention is not limited to the precise arrangements and instrumentalities shown, wherein:

FIG. 1 is a high level block diagram of an exemplary system for providing dynamic workload control and transition; and

FIG. 2 shows exemplary XML formatted workloads having varying loads.

100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255 256 257 258 259 260 261 262 263 264 265 266 267 268 269 270 271 272 273 274 275 276 277 278 279 280 281 282 283 284 285 286 287 288 289 290 291 292 293 294 295 296 297 298 299 300 301 302 303 304 305 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320 321 322 323 324 325 326 327 328 329 330 331 332 333 334 335 336 337 338 339 340 341 342 343 344 345 346 347 348 349 350 351 352 353 354 355 356 357 358 359 360 361 362 363 364 365 366 367 368 369 370 371 372 373 374 375 376 377 378 379 380 381 382 383 384 385 386 387 388 389 390 391 392 393 394 395 396 397 398 399 400 401 402 403 404 405 406 407 408 409 410 411 412 413 414 415 416 417 418 419 420 421 422 423 424 425 426 427 428 429 430 431 432 433 434 435 436 437 438 439 440 441 442 443 444 445 446 447 448 449 450 451 452 453 454 455 456 457 458 459 460 461 462 463 464 465 466 467 468 469 470 471 472 473 474 475 476 477 478 479 480 481 482 483 484 485 486 487 488 489 490 491 492 493 494 495 496 497 498 499 500 501 502 503 504 505 506 507 508 509 510 511 512 513 514 515 516 517 518 519 520 521 522 523 524 525 526 527 528 529 530 531 532 533 534 535 536 537 538 539 540 541 542 543 544 545 546 547 548 549 550 551 552 553 554 555 556 557 558 559 560 561 562 563 564 565 566 567 568 569 570 571 572 573 574 575 576 577 578 579 580 581 582 583 584 585 586 587 588 589 590 591 592 593 594 595 596 597 598 599 600 601 602 603 604 605 606 607 608 609 610 611 612 613 614 615 616 617 618 619 620 621 622 623 624 625 626 627 628 629 630 631 632 633 634 635 636 637 638 639 640 641 642 643 644 645 646 647 648 649 650 651 652 653 654 655 656 657 658 659 660 661 662 663 664 665 666 667 668 669 670 671 672 673 674 675 676 677 678 679 680 681 682 683 684 685 686 687 688 689 690 691 692 693 694 695 696 697 698 699 700 701 702 703 704 705 706 707 708 709 710 711 712 713 714 715 716 717 718 719 720 721 722 723 724 725 726 727 728 729 730 731 732 733 734 735 736 737 738 739 740 741 742 743 744 745 746 747 748 749 750 751 752 753 754 755 756 757 758 759 760 761 762 763 764 765 766 767 768 769 770 771 772 773 774 775 776 777 778 779 780 781 782 783 784 785 786 787 788 789 790 791 792 793 794 795 796 797 798 799 800 801 802 803 804 805 806 807 808 809 810 811 812 813 814 815 816 817 818 819 820 821 822 823 824 825 826 827 828 829 830 831 832 833 834 835 836 837 838 839 840 841 842 843 844 845 846 847 848 849 850 851 852 853 854 855 856 857 858 859 860 861 862 863 864 865 866 867 868 869 870 871 872 873 874 875 876 877 878 879 880 881 882 883 884 885 886 887 888 889 890 891 892 893 894 895 896 897 898 899 900 901 902 903 904 905 906 907 908 909 910 911 912 913 914 915 916 917 918 919 920 921 922 923 924 925 926 927 928 929 930 931 932 933 934 935 936 937 938 939 940 941 942 943 944 945 946 947 948 949 950 951 952 953 954 955 956 957 958 959 960 961 962 963 964 965 966 967 968 969 970 971 972 973 974 975 976 977 978 979 980 981 982 983 984 985 986 987 988 989 990 991 992 993 994 995 996 997 998 999 1000

DETAILED DESCRIPTION OF THE INVENTION

The present invention provides a method and system for dynamic workload transition. Dynamic workload transition includes switching between workloads having varying load requirements in order to ensure that an e-business system can perform optimally under different load conditions. In accordance with the invention, a servlet can be configured as a core (main) workload driver that dynamically monitors certain system parameters that are indicative of the current state of the system. A servlet can be a Java program that extends the functionality of a Web server, generating dynamic content and interacting with web clients using a request-response paradigm. In this case, the web clients can be simulated by a user driver which issues hypertext transfer protocol (HTTP) requests to the core workload driver for processing. Based on the monitored system parameters, the servlet can dynamically determine when a system overload condition and/or performance degradation is occurring.

Upon determining the existence of an overload condition, the servlet can diminish processing of the current workload by transitioning to lighter workload. This transition to a lightened load will not contribute to the system overload conditions and therefore can provide assistance in having the system return to an optimal operating state. Diminishing a workload means that the resources currently allocated to a workload is decreased. As an illustration of the diminishing of resources, the available

resources dedicated to WRKLD_005 can be diminished from 35% to 10% and the resources dedicated to workload WRKLD_001 can be increased from 65% to 90%.

FIG. 1 is a high level block diagram of an exemplary system for providing dynamic workload control and transition. Referring to FIG. 1, there is shown an application server 100, a core workload driver 110, a status driver 135, a monitor 140, and status indicator 145.

Application server 100 is a platform that can deliver application services to servers, databases, clients, applications, servlets and devices, which can be used to process e-business related transactions. Application servers are known in the art and depending on the vendor, can provide additional enhanced services to support specific vendee requirements. For example, some application servers can contain built in traffic flow management which can include traffic monitoring and control functions.

The core workload driver 110 can be a servlet which can be instantiated by the application server 100, and which can be configured to receive workload requests. The request can be initiated by a user driver. For example, core workload driver can be configured to receive and process workload requests such as WRKLD_005 105. The core workload driver 110 runs on the application server 100. The core workload driver 110 contains software logic that will ultimately execute the commands defined for the workload to be executed.

Status driver 135 can be a servlet which can be instantiated by the application server 100, and collects and reports information related to the overall system

performance. The system information can be collected from monitor 140. Monitor 140 can be a system management application or tool that can be used to collect system information. System management applications or tools are well known in the art and typically utilize agents to collect and report system information.

FIG. 2 shows exemplary XML formatted workloads having varying loads requirements. Referring to FIG. 2, exemplary workload definitions 200, 205, 210, 215 are shown. A workload definition contains the commands that define the tasks that are to be executed for a workload. Workload definitions 200, 205, 210, 215 can be defined in a configuration file that can be accessed by the core workload driver 110. Although the workload definitions illustrated in FIG. 2 utilize XML formatting, it should readily be understood by one skilled in the art that the use of XML formatted workload definitions in a configuration file is not intended to be a limitation on the system. Accordingly, other communication formats similar to XML and/or plain text, can be utilized to define the workloads.

The workload definitions 200, 205, 210, 215 have varying load requirements since the computing resources necessary for processing the commands for the tasks defined in the respective workloads vary in degree. Referring to FIG. 2, XML formatted workload definition (TEXTOUTPUT) 200 defines a workload which can provide text results to a user. XML formatted workload definition (IMAGEOUTPUT) 205 defines a workload that can provide image results to a user. The workload defined by workload definition 205 requires more data and multiple interactions with the application server 100, than is required by workload defined by the workload definition 200. This is mostly

due to the fact that the workload defined by workload definition 205 has commands that require image processing which utilizes more computing resources than is required for processing the text defined by the commands of workload definition 205.

Consequently, the computing resources necessary for processing the workload defined by workload definition 205 is greater than the computing resources necessary for processing the workload defined by workload definition 200. The load on the system due to the commands in workload 205 is, therefore, greater than the load due to the commands on workload 200.

Workload 210 can be used to execute a query (SHORTQUERY) which permits fetching up to 10 rows of data. Workload 215 can be used to execute a query (LONGQUERY) which can permit up to 3000 rows of data to be returned. As a result, workload 215 requires significantly more computing resources than workload 210, since workload 215 permits up to 3000 rows of data to be fetched.

Based on the workload definitions, the core driver 110 can assess overall system load and accordingly determine which workload should be processed. Subsequent to receiving a workload request, for example WRKLD_005 105, the core workload driver 110 will attempt to execute the command(s) defined by the workload. The core workload driver 110 contains software logic that will ultimately execute the commands defined for the workload being executed. Factors including the priority of the requesting user, the current load on the system due to other commands being executed and the amount of processing resources that will be required for any requesting and new workloads will be considered by the core workload driver 110. Monitor 140 can monitor

factors such as the system load. Monitored data related to the system load can be collected by the status driver 135 and reported to the core workload driver 110. The monitored data relating to the system load can include, but is not limited to, parameters such as CPU utilization, memory usage, page Ins, disk I/O and run queue. These parameters can either be passed to the core workload driver 110 by the status monitor 135 or made accessible to the core workload driver 110.

The software logic contained in the core workload driver 110 can assess the data from the status driver 135, along with other data related to the factors previously described, and will determine whether it is appropriate to transition from one workload to another. For example, if the status driver 135 reports that the system is 75% utilized, and a request from a user includes a database query requiring 1 million rows of data, the core workload driver 110 will make a determination that the request cannot be immediately executed. Consequently, the software logic can transition the workload request and perform alternate lighter workloads which require less processing resources. In this case, the software logic in the core workload driver 110 could make a determination that the first 100,000 rows of data could be obtained and sent. The core workload driver 110 could subsequently send a message to the requestor, requiring the requestor to make a subsequent request to acquire the remainder of the requested data. In another aspect of the invention, a new request can be made in order to process the original 1 million rows.

As an illustration and with reference to FIG. 1, the status driver 135 can collect system data from monitor 140. The status driver can report the collected system data

to the core workload driver 110. A status indicator 145 depicts the collected system data that signifies the system load. Status indicator 145 indicates that the system is approximately 65% utilized. The core workload driver 110 receives workload request 105 (WRKLD_005). The core workload driver 110 determines that there is sufficient resources available to process two requests for WRKLD_005 105A and 105B as indicated by section 120 of the status indicator 145. However, there is insufficient resources to process a subsequent request portion 105C of WRKLD_005 as indicated by 125 of the status indicator 145. Consequently, the core workload driver 110 transitions any requests for WRKLD_005 to lighter workloads (WRKLD_001), as depicted by 115A and 115B. WRKLD_001 can be a lighter workload or it can be a newly requested workload. By defining the workloads in a system so that they have varying workloads, it is possible to easily transition between heavier and lighter workloads whenever there is an overload condition.

Whenever the status driver 135 reports that there is sufficient processing resources available as illustrated by section 130 of the status indicator, the core workload driver 110 will process requests for WRKLD_005 as depicted in 105C . It should be recognized that in transitioning to a lighter workload, the amount of resources being utilized are diminished. When the unavailability of system resources dictate that a transition be made to a lighter workload, the heavier workload can be processed as normal. Whenever adequate resources become available an unrelated new load can be processed.

The present invention can be realized in hardware, software, or a combination of

hardware and software. A method and apparatus for providing dynamic workload transition according to the present invention can be realized in a centralized fashion in one computer system, or in a distributed fashion where different elements are spread across several interconnected computer systems. Any kind of computer system, or other apparatus adapted for carrying out the methods described herein, is suited. A typical combination of hardware and software could be a general purpose computer system with a computer program that, when being loaded and executed, controls the computer system such that it carries out the methods described herein. The present invention can also be embedded in a computer program product, which comprises all the features enabling the implementation of the methods described herein, and which - when loaded in a computer system is able to carry out these methods.

Computer program or application in the present context means any expression, in any language, code or notation, of a set of instructions intended to cause a system having an information processing capability to perform a particular function either directly or after either or both of the following a) conversion to another language, code or notation; b) reproduction in a different material form.